МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра прикладной математики и информатики

ОТЧЁТ

по лабораторной работе №7 «Руководство программиста»

по дисциплине

Структуры данных

РУКОВОДИТЕЛЬ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Санников А.Н.

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Зырянов Е.А.

22-ПМ-1

Работа защищена «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_

С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Задание на лабораторную работу:*** Написать документацию «Руководство программиста»

***Оглавление***

**1. Общие сведения**

1.1. Сокращения

1.2. Назначение

1.3. Функциональные возможности

1.4. Условия применения

1.5. Требования к квалификации разработчика

**2. Прикладное решение и его компоненты**

2.1. Понятие прикладного решения

2.2. Требования к прикладному решению

2.2.1. Базовые требования

2.2.2. Требования к входным данным

2.2.3. Требования к алгоритмам расчета данных

2.2.4. Требования к отчетам и набору каждого из них

2.2.5. Требования к разграничению доступа и информационной безопасности

2.3. Компоненты прикладного решения

2.3.1. Структура базы данных

2.3.2. Загрузка исходных данных

2.3.3. Организация пользовательского интерфейса

2.3.4. Работа пользователя с данными

2.3.5. Обработка и расчет данных

2.3.6. Отображение данных

2.3.7. Средства автоматической загрузки и обработки данных.

2.4. Разграничение прав доступа в прикладном решении

2.4.1. Роли пользователей

2.4.2. Профили пользователей

2.5. Жизненный цикл прикладного решения

**3. Реализация прикладного решения**

3.1. **Постановка задачи. Определение требований к прикладному решению**

3.2. **Проектирование прикладного решения**

**3.2.1. Проектирование структуры таблиц и процедур загрузки данных**

3.2.2. **Проектирование интерфейса пользователя**

**3.2.3. Проектирование процедур расчета и отчетов**

3.3. **Реализация прикладного решения**

**3.3.1. Создание процедур загрузки данных**

**3.3.2. Создание справочников и визуальных форм для них**

**3.3.3. Создание процедур расчета на языке хранимых процедур базы данных**

**3.3.4. Создание групп пользователей и создание главного меню для них**

**3.3.5. Создание процедуры автоматического обновления (загрузки) данных**

**3.4.Развертывание прикладного решения**

**3.5. Тестирование прикладного решения**

**3.6. Усложнение задачи**

**3.7. Финальный этап внесения доработок в прикладное решение**

*****1. Общие сведения*****

*1.1.* ***Сокращения***

SFML: Simple and Fast Multimedia Library. Кросс-платформенная библиотека, предоставляющая простой интерфейс к различным компонентам ПК.

AI: Artificial Intelligence. Искусственный интеллект.

*1.2.****Назначение***

Код представляет собой игру “Квадраты”, в которой игрок соревнуется с компьютером. Это классическая настольная игра, цель которой - захватить как можно больше квадратов на поле.

*1.3.* ***Функциональные возможности***

Игра предоставляет интерактивный графический интерфейс для игры “Квадраты”. Интерфейс позволяет игроку легко взаимодействовать с игрой, делая ходы и видя результаты своих действий.

Игра включает в себя AI, который использует алгоритм минимакса для определения оптимального хода. Это делает игру более интересной и вызывающей, поскольку AI стремится сделать наилучший возможный ход.

*1.4.* ***Условия применения***

Для работы кода требуется компилятор C++ и библиотека SFML. Это означает, что код может быть запущен на любой системе, которая поддерживает эти инструменты.

*1.5.* ***Требования к квалификации разработчика***

Разработчик должен иметь знания в области C++, а также понимание библиотеки SFML и алгоритма минимакса с альфа-бета отсечением. Это обеспечит возможность разработчика понять, изменить и расширить код при необходимости.

*****2. Прикладное решение и его компоненты*****

*2.1.* ***Понятие прикладного решения***

Код состоит из нескольких функций и структур, которые вместе образуют игру “Квадраты”. Это прикладное решение, которое решает конкретную задачу - предоставление интерактивной игры “Квадраты”.

*2.2.* ***Требования к прикладному решению***

*2.2.1.* ***Базовые требования***

Код должен успешно компилироваться и запускаться. Это обеспечивает основу для функционирования игры.

Игра должна предоставлять интерактивный графический интерфейс для игры “Квадраты”. Это обеспечивает удобство и простоту использования для пользователя.

*2.2.2.* ***Требования к входным данным***

Входные данные получаются от действий пользователя в графическом интерфейсе. Это включает в себя ходы, которые делает игрок во время игры.

*2.2.3.* ***Требования к алгоритмам расчета данных***

Алгоритмы должны корректно обрабатывать ходы игрока и AI. Это включает в себя определение, кто сделал ход, обновление состояния игры и определение, когда игра закончена.

*2.2.4.* ***Требования к отчетам и набору каждого из них***

Игра должна корректно отображать текущее состояние игрового поля. Это включает в себя отображение квадратов, которые были захвачены игроком или AI.

*2.2.5.* ***Требования к разграничению доступа и информационной безопасности***

Не применимо, поскольку код не включает в себя функции для разграничения доступа или обеспечения информационной безопасности.

*2.3.* ***Компоненты прикладного решения***

*2.3.1.* ***Структура базы данных***

Не применимо, поскольку код не использует базу данных.

*2.3.2.* ***Загрузка исходных данных***

Не применимо, поскольку код не загружает внешние данные.

*2.3.3.* ***Организация пользовательского интерфейса***

Интерфейс организован с использованием библиотеки SFML. Это обеспечивает графический интерфейс, который позволяет пользователю взаимодействовать с игрой.

*2.3.4.* ***Работа пользователя с данными***

Пользователь взаимодействует с игрой через графический интерфейс, делая ходы в игре “Квадраты”. Это включает в себя выбор хода и просмотр результатов хода.

*2.3.5.* ***Обработка и расчет данных***

Данные обрабатываются с помощью функций, которые реализуют логику игры и алгоритм минимакса. Это обеспечивает корректное обновление состояния игры после каждого хода.

*2.3.6.* ***Отображение данных***

Данные отображаются в графическом интерфейсе игры. Это включает в себя отображение текущего состояния игрового поля и ходов, сделанных игроком и AI.

*2.3.7.* ***Средства автоматической загрузки и обработки данных***

Не применимо, поскольку код не включает в себя функции для автоматической загрузки или обработки данных.

*2.4.* ***Разграничение прав доступа в прикладном решении***

Не применимо, поскольку код не включает в себя функции для разграничения прав доступа.

*2.5.* ***Жизненный цикл прикладного решения***

Жизненный цикл прикладного решения начинается с запуска программы и завершается при закрытии игры. В течение этого цикла игрок и AI делают ходы, и игра обновляет свое состояние в соответствии с этими ходами.

*****3. Пример прикладного решения*****

Данный код является примером прикладного решения. Он демонстрирует, как можно создать игру “Квадраты” на C++ с использованием библиотеки SFML и алгоритма минимакса с альфа-бета отсечением для AI.

*3.1.* ***Постановка задачи. Определение требований к прикладному решению***

Задача состоит в создании игры “Квадраты” на C++ с использованием библиотеки SFML для графического интерфейса и алгоритма минимакса для AI. Требования включают корректную работу игры, интерактивный графический интерфейс и эффективный AI.

*3.2.* ***Проектирование прикладного решения***

*3.2.1.* ***Проектирование структуры таблиц и процедур загрузки данных***

В данном случае, структура данных представлена в виде структуры *field*, которая содержит игровое поле и методы для его обработки. Загрузка данных происходит через взаимодействие пользователя с графическим интерфейсом.

*3.2.2.* ***Проектирование интерфейса пользователя***

Интерфейс пользователя реализован с использованием библиотеки SFML. Он представляет собой окно, в котором отображается игровое поле, и обрабатывает ввод пользователя через мышь.

*3.2.3.* ***Проектирование процедур расчета и отчетов***

Процедуры расчета включают функции, которые реализуют логику игры и алгоритм минимакса. Отчеты представлены в виде отображения текущего состояния игрового поля.

*3.3.* ***Реализация прикладного решения***

*3.3.1.* ***Создание процедур загрузки данных***

Загрузка данных происходит через взаимодействие пользователя с графическим интерфейсом.

*3.3.2.* ***Создание справочников и визуальных форм для них***

Не применимо, поскольку код не включает в себя функции для создания справочников или визуальных форм.

*3.3.3.* ***Создание процедур расчета на языке хранимых процедур базы данных***

Не применимо, поскольку код не использует базу данных. Вместо этого, все расчеты выполняются в оперативной памяти.

*3.3.4.* ***Создание групп пользователей и создание главного меню для них***

Не применимо, поскольку код не включает в себя функции для создания групп пользователей или главного меню.

*3.3.5.* ***Создание процедуры автоматического обновления (загрузки) данных***

Не применимо, поскольку код не включает в себя функции для автоматического обновления или загрузки данных.

*3.4.****Развертывание прикладного решения***

Развертывание прикладного решения включает в себя компиляцию кода и запуск исполняемого файла на целевой системе.

*3.5.* ***Тестирование прикладного решения***

Тестирование прикладного решения может включать в себя проверку корректности работы игры, включая правильность ходов и определение победителя.

*3.6.* ***Усложнение задачи***

Усложнение задачи может включать в себя добавление новых функций в игру, таких как различные уровни сложности или новые варианты игрового поля.

*3.7.* ***Финальный этап внесения доработок в прикладное решение***

Финальный этап внесения доработок в прикладное решение может включать в себя исправление ошибок, оптимизацию производительности и улучшение пользовательского интерфейса.

**Вывод:** Научились писать документацию «Руководство программиста».